



УТВЕРЖДАЮ

Проректор по учебной работе БГПУ

В.В. Шлыков

130 10. 2014 г.

Регистрационный № УД- 24-04-н 36-2014 р.

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ И МЕТОДЫ АЛГОРИТМИЗАЦИИ

Учебная программа учреждения высшего образования по учебной дисциплине
для специальности:

1-02 05 01 Математика и информатика

Факультет **математический**

Кафедра **прикладной математики и информатики**

Курс 1 – 2

Семестр 2 – 4

Лекции 58

Экзамен 4

Практические (семинарские)
занятия _____

Зачет 2, 3
(семестр)

Лабораторные
занятия 88 + 4 УСРС

Курсовая работа _____

Аудиторных часов по
учебной дисциплине 150

Контрольная работа _____

Всего часов по
учебной дисциплине 318

Форма получения
высшего образования дневная

Составители С.И. Зенько, канд. пед. наук, доцент,
Н.Н. Нарейко, старший преподаватель,
В.В. Пенкрат, старший преподаватель,
А.З. Кутыш, преподаватель

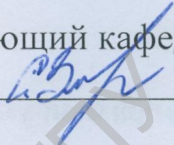
2014 г.

Учебная программа учреждения высшего образования составлена на основе учебной программы по учебной дисциплине «Технологии программирования и методы алгоритмизации», регистрационный № ТД-А. 501/тип.

Рассмотрена и рекомендована к утверждению кафедрой прикладной математики и информатики.

18 сентября 2014 г., протокол № 2

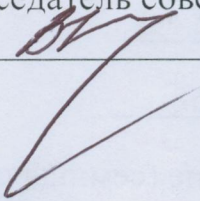
Заведующий кафедрой


С.И. Зенько

Одобрена и рекомендована к утверждению Советом математического факультета БГПУ.

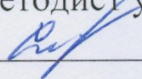
24 сентября 2014 г., протокол № 2

Председатель совета факультета


В.А. Шилинец

Оформление учебной программы и сопровождающих её материалов действующим требованиям Министерства образования Республики Беларусь соответствует

Методист учебно-методического управления БГПУ


С.А. Стародуб

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина «Технологии программирования и методы алгоритмизации» предназначена для студентов, обучающихся по специальности 1–02 05 01 «Математика и информатика». Типовая учебная программа по учебной дисциплине «Технологии программирования и методы алгоритмизации» составлена в соответствии с требованиями образовательного стандарта.

Обучение учебной дисциплине проходит в рамках организации лекционных и лабораторных занятий. При чтении лекций особое внимание следует уделять использованию мультимедийных технологий для демонстрации особенностей и возможностей изучаемых языков и технологий программирования.

Лабораторные занятия направлены на закрепление лекционного теоретического материала и на формирование навыков практического использования полученных знаний при выполнении конкретных заданий. Организация лабораторных работ предполагает использование личностно-ориентированных методов обучения, что способствует развитию индивидуально-творческих способностей каждого студента и приобретению навыков самостоятельной работы.

Содержание и формы контролируемой самостоятельной работы студентов разрабатываются кафедрой в соответствии с целями и задачами специалиста. Для управления самостоятельной работой студентов рекомендуется использовать электронные средства обучения.

Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

На протяжении истории развития информационных технологий эволюция языков программирования означала изменение вычислительной среды, способа мышления и самого подхода к программированию (парадигмы программирования). Таким образом развивались технологии программирования от структурного к объектно-ориентированному и компонентному программированию.

Практикоориентированность учебной дисциплины «Технологии программирования и методы алгоритмизации» очевидна. В настоящее время происходит быстрое развитие ИТ-технологий и, соответственно, увеличивается спрос на ИТ-специалистов. Будущему преподавателю информатики недостаточно знаний только одного языка программирования. Необходимо также знать основные принципы современных технологий программирования. Это позволяет быть конкурентоспособным и мобильным как в системе образования, так и в профессиональной сфере в целом.

Целью преподавания учебной дисциплины является формирование профессиональных компетенций учителя математики и информатики в области технологий программирования и методов алгоритмизации.

Основными **задачами** учебной дисциплины являются:

- формирование знаний теоретических основ алгоритмизации;
- овладение понятийным аппаратом языков и технологий программирования;
- изучение основных принципов структурного, объектно-ориентированного, обобщенного и компонентно-ориентированного программирования;
- приобретение умений разработки алгоритмов, классов, обобщенных наборов данных, компонентов и программ их реализации.

Тематика учебной дисциплины «Технологии программирования и методы алгоритмизации» способствует выявлению и развитию у студентов системных подходов к конструированию объектных моделей из разных предметных областей, логического и объектно-ориентированного мышления, подготовке студентов к активной профессиональной деятельности в условиях современного общества.

Содержание учебной дисциплины делится на несколько логических частей (разделов). Первые два раздела являются пропедевтикой к изучению основ современных технологий программирования. Так, первый раздел учебной дисциплины предназначен для знакомства студентов с основными понятиями теории и практики алгоритмизации и языков программирования. Раскрываются парадигмы программирования. Вторым разделом учебной дисциплины посвящен структурному программированию, методам алгоритмизации (поиск в массиве, упорядочивание элементов массива, циклические перестановки элементов массива) и их реализации.

Третий раздел учебной дисциплины посвящен изучению принципов и средств реализации современных парадигм программирования (объектно-ориентированного, обобщенного, компонентно-ориентированного программирования и др.), формированию умений визуального проектирования графического интерфейса приложений, приобретению опыта разработки Windows-приложений прикладного характера.

При изучении учебной дисциплины «Технологии программирования и методы алгоритмизации» рекомендуется использовать различные современные методики преподавания в учреждениях высшего образования, в частности, последовательного обучения, параллельного изложения некоторых тем, на основе взаимосвязанного обучения и др.

В результате изучения учебной дисциплины студент должен

знать:

- специальную терминологию современных технологий программирования;
- основные структуры данных и базовые структуры алгоритмов;
- методы алгоритмизации;
- сущность и особенности структурно-модульного и объектно-ориентированного программирования;
- основные приемы (визуальное проектирование, событийное программирование и др.) использования современных технологий программирования при разработке приложений.

В результате изучения учебной дисциплины студент должен

уметь:

- работать в современных средах программирования;
- использовать основные конструкции языка программирования высокого уровня при проектировании и отладке алгоритмов;
- применять методы алгоритмизации при разработке программ на языке высокого уровня;
- разрабатывать объектные модели в различных предметных областях;
- создавать приложения прикладного характера с помощью современных технологий программирования.

В результате изучения учебной дисциплины студент должен

владеть:

- способами проектирования алгоритмов и их реализацией с помощью современных средств программирования;
- приемами разработки приложений на основе объектных моделей и их тестирования;
- методами и средствами современных парадигм программирования: объектно-ориентированного, обобщенного, компонентно-ориентированного и др.

Программа учебной дисциплины «Технологии программирования и методы алгоритмизации» рассчитана на 318 учебных часов, из них аудиторных 150: лекций – 58 часов, лабораторных работ – 92 часа. Место учебной дисциплины определено учебным планом во 2 – 4 семестрах. Примерное распределение учебных часов по семестрам: 2 семестр – 68 часов (28 часов – лекции, 40 часов – лабораторные занятия); 3 семестр – 42 часа (18 часов – лекции, 24 часа – лабораторные занятия); 4 семестр – 40 часов (12 часов – лекции, 28 часов – лабораторные занятия).

Рекомендуемые формы итогового контроля – зачет (2 и 3 семестры), экзамен (4 семестр).

Освоение системы знаний и умений по учебной дисциплине «Технологии программирования и методы алгоритмизации» позволит студентам продуктивно осваивать учебный материал по таким учебным дисциплинам, как «Информационные системы и сети», «Методика преподавания информатики», «Практикум по решению задач по информатике», «Информационные технологии в образовании».

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

РАЗДЕЛ I. ОСНОВЫ ТЕОРИИ И ПРАКТИКИ АЛГОРИТМИЗАЦИИ И ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Тема 1.1. Основы алгоритмизации

Этапы решения задач на компьютере.

Понятие алгоритма. Свойства алгоритма. Способы представления алгоритмов.

Блок-схемы алгоритмов, основные элементы. Базовые структуры.

Тема 1.2. Основы теории языков программирования

Понятие языка программирования. Область применения языков программирования. Критерии эффективности языков программирования.

Пути эволюции языков и технологий программирования. Парадигмы программирования. Виды языков программирования. Парадигма структурного программирования.

Тема 1.3. Системы и среды программирования

Понятия систем и сред программирования. Назначение, основные функции и состав системы программирования.

РАЗДЕЛ II. СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ

Тема 2.1. Основные элементы языка программирования

Основные понятия языка программирования: символ, слова, величина.

Скалярные (простые) типы данных: перечисляемый, целый, вещественный, логический, символьный и диапазонный.

Операции и выражения. Унарные и бинарные операции. Приоритетность операций. Арифметические и логические выражения. Совместимость типов операндов выражения.

Основные арифметические функции.

Структура программы.

Тема 2.2. Операторы языка программирования. Базовые алгоритмические конструкции

Оператор присваивания. Комментарии в программе. Процедуры ввода и вывода. Форматы вывода числовых данных.

Понятие структурного оператора.

Алгоритмическая конструкция «Следование». Составной оператор.

Алгоритмическая конструкция «Ветвление». Оператор альтернативы (условный). Полная и неполная формы оператора альтернативы. Оператор выбора.

Алгоритмическая конструкция «Повторение». Понятия «цикл», «тело цикла», «условие цикла». Циклы с определяемым количеством повторений. Оператор цикла *while*. Оператор цикла *repeat*. Цикл с известным количеством повторений. Оператор цикла *for*. Вложенные циклы.

Тема 2.3. Подпрограммы

Понятие подпрограммы. Виды подпрограмм. Локальные и глобальные переменные.

Процедуры пользователя. Структура процедуры пользователя. Организация вызова процедуры пользователя. Типы параметров (фактические и формальные, параметры-значения и параметры-переменные). Виды процедур пользователя: процедуры без параметров, процедуры с параметрами-значениями, процедуры с параметрами-значениями и параметрами-переменными.

Функции пользователя. Структура функции пользователя. Организация вызова функции пользователя. Отличия функции пользователя от процедуры пользователя.

Понятие итерации. Понятие рекурсии. Рекурсивные подпрограммы.

Тема 2.4. Структурированные (составные, сложные) типы данных. Методы алгоритмизации

Понятие строковой величины. Объявление строковых величин. Операции над строковыми величинами. Процедуры и функции работы со строковыми величинами.

Понятие массива. Одномерные и двумерные массивы: описание массивов, способы ввода и вывода элементов массива.

Методы алгоритмизации: поиск в массиве, упорядочивание (сортировка) элементов массива, циклические перестановки элементов массива.

Методы поиска элемента массива: поиск перебором, бинарный поиск и др. Поиск элементов в одномерных и двумерных массивах с заданными свойствами (свойства элементов, лежащих на (под, над) главной и побочной диагоналями квадратной матрицы и др.).

Методы упорядочивания элементов массива: сортировка выбором, сортировка обменом (пузырьковая) и др.

Циклические перестановки элементов массива.

Анализ алгоритмов. Сложность алгоритмов.

Понятие множества. Описание множеств. Операции над множествами: объединение, пересечение, разность, операции сравнения, операция *in*.

Понятия записи, поля записи. Описание записей. Массивы записей. Оператор *with*.

Понятие файлов. Виды файлов. Типизированные файлы. Работа с типизированными файлами: создание файла, использование данных из файла, дополнение файла новыми данными.

Тема 2.5. Основы программирования графики и звука

Основы работы с графикой. Модуль *Graph*. Графические примитивы. Работа с цветом. Работа с пером и кистью. Действия со шрифтами. Имитация движения графических объектов. Построение графиков элементарных функций. Деловая графика (столбчатые и круговые диаграммы).

Основы работы со звуком. Модуль *Sounds*. Процедуры и функции для работы со звуком.

РАЗДЕЛ III. ОСНОВЫ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ

Тема 3.1. Парадигма объектно-ориентированного программирования.

Платформа .NET

Объектно-ориентированные языки и среды программирования. Философия и технология .NET. Компоненты платформы .NET. Инфраструктура .NET Framework, строительные блоки .NET – CLR, CTS, CLS. Базовые классы .NET.

Модель выполнения приложений в среде .NET Framework. Промежуточный язык MSIL (*Microsoft Intermediate Language*).

Тема 3.2. Основы объектно-ориентированного и компонентно-ориентированного языка программирования C#

Синтаксис языка.

Встроенные типы и системные типы. Пространство имен *System*. Директива *using System*. Встроенная утилита *Object Browser*, просмотр содержимого пространства имен *System*.

Класс *System.Console*. Операции ввода и вывода. Чтение строки символов. Преобразование строки символов в число, метод *Parse()*. Формат вывода. Класс *System.Math*, его методы и константы.

Операции, приоритеты операций. Операторы и выражения. Преобразование типов в выражениях.

Массивы. Класс *System.Random*, применение его методов для создания массива случайных чисел. Класс *System.Array*, его методы и свойство *Length*. Цикл *foreach*, перебор элементов массивов.

Структуры. Создание структур. Работа со структурами.

Тема 3.3. Объектно-ориентированное проектирование и программирование

Понятие объектно-ориентированного программирования. Объектные модели, принципы их построения, особенности реализации.

Классы в C#. Форма определения класса. Члены класса. Спецификаторы доступа к членам класса.

Конструкторы. Параметризованные конструкторы. Назначение оператора *new*. Создание экземпляра класса.

Типы значений. Ссылочные типы. Динамическое использование памяти: стеки и кучи. Сбор «мусора» и использование деструкторов.

Определение пользовательских методов класса. Спецификаторы доступа к методам класса. Модификаторы *ref*, *out*, *params* параметров методов.

Возвращение методами объектов класса. Возвращение методами массивов.

Общие (статические) члены класса. Доступ к общим членам класса. Ограничения на *static*-методы.

Область видимости и время существования переменных. Преобразование и приведение типов.

Обработка исключительных ситуаций.

Тема 3.4. Принципы объектно-ориентированного программирования

Перегрузка методов и конструкторов как механизм реализации полиморфизма.

Перегрузка операторов. Перегрузка бинарных, унарных и операторов отношений.

Инкапсуляция и свойства. Формат записи свойств и правила их использования.

Индексаторы. Одномерные индексаторы.

Наследование, базовые и производные классы. Доступы к членам базовых классов. Конструкторы и наследование. Виртуальные методы. Абстрактные классы. Переопределение виртуальных и абстрактных методов переопределение как механизм реализации полиморфизма.

Интерфейсы и множественное наследование. Определение. Реализация интерфейсов.

Тема 3.5. Принципы обобщенного программирования и их реализация

Понятие обобщенного программирования. Принципы обобщенного программирования (абстрагирование, иерархия, типизация).

Коллекции и наборы. Обзор коллекций. Интерфейсы коллекций. Классы коллекций общего назначения.

Классы *ArrayList*, *Queue*, *Stack*, другие типы наборов данных из пространства имен *System.Collections*. Методы работы с различными структурами данных.

Динамическая идентификация типов. Проверка типа с помощью ключевого слова *is*. Использование операторов *as*, *typeof*.

Обобщения. Разработка обобщенных наборов, методов и классов.

Тема 3.6. Основы компонентно-ориентированного программирования

Принципы работы *Windows*-приложений.

Делегаты. Назначение делегатов. События и сообщения, их обработка. Объявление и генерация событий. Реализация обработчиков событий.

Разработка объектной модели приложения.

Графический интерфейс приложений. Технология *Windows Forms*. Пространство имен *System.WinForms*, класс *Form*. Добавление форм к проекту. Методы и события формы.

Элементы управления. Создание обработчиков событий элементов управления.

Понятие компонента и компонентной модели. Разработка программных компонентов. Приложение-сервер, приложение-контейнер (клиент). Интерфейс *System.ComponentModel.IComponent* и класс *System.ComponentModel.Component*.

Использование собственных программных компонентов в *Windows*-приложениях прикладного характера.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов						Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Управляемая самостоятельная работа	Иное	
1	2	3	4	5	6	7	8	9
1	Основы теории и практики алгоритмизации и языков программирования	4						
1.1	Основы алгоритмизации	2						
1.1.1	Основы алгоритмизации 1. Этапы решения задач на компьютере. 2. Понятие алгоритма. Свойства алгоритма. Способы представления алгоритмов. 3. Блок-схемы алгоритмов, основные элементы. Базовые структуры.	2						Устный опрос.
1.2	Основы теории языков программирования	1						
1.2.1	Основы теории языков программирования 1. Понятие языка программирования. Область применения языков программирования. 2. Критерии эффективности языков программирования. 3. Пути эволюции языков и технологий программирования. Парадигмы программирования. Виды языков программирования. 4. Парадигма структурного программирования.	1						Устный опрос.

1	2	3	4	5	6	7	8	9
1.3	Системы и среды программирования	1						
1.3.1	Системы и среды программирования 1. Понятия систем и сред программирования. 2. Назначение, основные функции и состав системы программирования.	1						Устный опрос.
2	Структурное программирование	24			36	4		
2.1.	Основные элементы языка программирования	2			2			
2.1.1.	Основные элементы языка программирования 1. Основные понятия языка программирования: символ, слова, величина. 2. Скалярные (простые) типы данных: перечисляемый, целый, вещественный, логический, символьный и диапазонный. 3. Операции и выражения. Унарные и бинарные операции. Приоритетность операций. Арифметические и логические выражения. Совместимость типов операндов выражения. 4. Основные арифметические функции. 5. Структура программы.	2						Устный опрос.
2.1.2.	Типы переменных, выражения Указание типов переменных в программе. Определение типов констант. Арифметические выражения и их типы. Логические выражения. Запись математических выражений в программе.				2			Проверка лабораторной работы
2.2	Операторы языка программирования. Базовые алгоритмические конструкции	6			8	2		
2.2.1	Операторы языка программирования 1. Оператор присваивания. Комментарии в программе. 2. Процедуры ввода и вывода. 3. Форматы вывода числовых данных. 4. Понятие структурного оператора. 5. Алгоритмическая конструкция «Следование». Составной оператор.	2						Устный опрос.

1	2	3	4	5	6	7	8	9
2.2.2	<p>Составление программ с использованием алгоритмической конструкции «Следование»</p> <p>Составление блок-схем решения задач с использованием алгоритмической конструкции «Следование». Объявление констант и переменных в программе. Процедуры ввода и вывода в программе и их геометрическое представление. Оператор присваивания в программе и его геометрическое представление.</p>				2			Проверка лабораторной работы
2.2.3	<p>Алгоритмическая конструкция «Ветвление»</p> <p>1. Алгоритмическая конструкция «Ветвление». 2. Оператор альтернативы (условный). 3. Полная и неполная формы оператора альтернативы. 4. Оператор выбора.</p>	2						Устный опрос.
2.2.4	<p>Составление программ с использованием алгоритмической конструкции «Ветвление»</p> <p>Полная и неполная формы оператора альтернативы в программах и их геометрическое представление. Полная и неполная формы оператора выбора в программах и их геометрическое представление.</p>				2			Проверка лабораторной работы
2.2.5	<p>Алгоритмическая конструкция «Повторение»</p> <p>1. Алгоритмическая конструкция «Повторение». 2. Понятия «цикл», «тело цикла», «условие цикла». 3. Циклы с определяемым количеством повторений. Оператор цикла while. Оператор цикла repeat. 4. Цикл с известным количеством повторений. Оператор цикла for. 5. Вложенные циклы.</p>	2						Устный опрос.

1	2	3	4	5	6	7	8	9
2.2.6	Составление программ с использованием алгоритмической конструкции «Повторение» Оператор цикла с предусловием в программах и его геометрическое представление в блок-схемах. Оператор цикла с постусловием в программах и его геометрическое представление в блок-схемах. Составление программ с использованием операторов цикла.				2			Проверка лабораторной работы
	Оператор цикла с параметром в программах и две формы его представления в блок-схемах. Их представление в виде блок-схем.					2		Проверка самостоятельной работы
2.2.7	Составление программа с использованием вложенных циклов Понятие внешнего и внутреннего циклов и их представление в виде блок-схем. Количество повторений для внешнего и внутреннего циклов. Сочетание разных операторов циклов в программах с использованием вложенных циклов.				2			Проверка лабораторной работы
2.3	Подпрограммы	4			6			
2.3.1	Подпрограммы 1. Понятие подпрограммы. 2. Виды подпрограмм. 3. Локальные и глобальные переменные.	1						Устный опрос.
2.3.2	Процедуры и функции пользователя 1. Процедуры пользователя. Структура процедуры пользователя. Организация вызова процедуры пользователя. 2. Типы параметров (фактические и формальные, параметры-значения и параметры-переменные). 3. Виды процедур пользователя: процедуры без параметров, процедуры с параметрами-значениями, процедуры с параметрами-значениями и параметрами-переменными.	2						Устный опрос.

1	2	3	4	5	6	7	8	9
	4. Функции пользователя. Структура функции пользователя. Организация вызова функции пользователя. 5. Отличия функции пользователя от процедуры пользователя.							
2.3.3	Составление программ с использованием процедур пользователя Составление программ с использованием процедуры пользователя без параметров. Составление программ с использованием процедуры пользователя без параметров-переменных. Составление программ с использованием процедуры пользователя общего вида.				2			Проверка лабораторной работы
2.3.4	Составление программ с использованием функций пользователя Составление программ с использованием функций пользователя без глобальных параметров. Составление программ с использованием функций пользователя с глобальными параметрами. Составление программ с использованием функций пользователя общего вида				2			Проверка лабораторной работы
2.3.5	Понятия итерации и рекурсии 1. Понятие итерации. 2. Понятие рекурсии. Рекурсивные подпрограммы.	1						Устный опрос.
2.3.6	Рекурсивные подпрограммы Составление программ с использованием рекурсивных функций пользователя. Составление программ с использованием рекурсивных процедур пользователя.				2			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
2.4	Структурированные (составные, сложные) типы данных. Методы алгоритмизации	10			16	2		
2.4.1	Строковые величины 1. Понятие строковой величины. 2. Объявление строковых величин. 3. Операции над строковыми величинами. 4. Процедуры и функции работы со строковыми величинами.	2						Устный опрос
2.4.2	Составление программ с использованием строковых величин Составление программ с использованием процедур и функций работы с отдельными словами. Составление программ для обработки текстовых предложений. Составление программ на основе перевода строковых переменных в числовые и наоборот.				2			Проверка лабораторной работы
2.4.3	Массивы 1. Понятие массива. 2. Одномерные и двумерные массивы: описание массивов, способы ввода и вывода элементов массива. 3. Главная и побочная диагонали двумерного массива.	1						Устный опрос
2.4.4	Составление программ с использованием массивов Ввод и вывод элементов одномерного массива. Составление программ с использованием одномерных массивов. Ввод и вывод элементов двумерного массива. Составление программ с использованием свойств главной и побочной диагоналей двумерных массивов.				2			Проверка лабораторной работы
2.4.5	Методы алгоритмизации 1. Поиск в массиве. 2. Упорядочивание (сортировка) элементов массива. 3. Циклические перестановки элементов массива: сущность.	1						Устный опрос

1	2	3	4	5	6	7	8	9
2.4.6	Методы поиска элемента массива 1. Поиск перебором. 2. Бинарный поиск. 3. Поиск элементов в одномерных и двумерных массивах с заданными свойствами: свойства элементов, лежащих на (под, над) главной и побочной диагоналями квадратной матрицы.	1						Устный опрос
2.4.7	Составление программ с использованием методов поиска в массивах Методы поиска наименьшего и наибольшего элемента в одномерном массиве. Поиск элементов с заданными свойствами в одномерном массиве. Поиск наименьшего и наибольшего элементов по строкам и столбцам в двумерном массиве. Поиск элементов с заданными свойствами на главной и побочной диагоналях двумерного массива.				2			Проверка лабораторной работы
2.4.8	Методы упорядочивания и перестановки элементов массива 1. Сортировка выбором. 2. Сортировка обменом. 3. Циклические перестановки элементов массива.	1						Устный опрос
2.4.9	Составление программ с использованием методов сортировки элементов массива Составление программ с использованием методов сортировки элементов одномерного массива. Составление программ с использованием сортировки элементов двумерного массива.				2			Проверка лабораторной работы
2.4.10	Анализ алгоритмов 1. Сущность анализа алгоритмов. 2. Сложность алгоритмов.				2			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
2.4.11	Множества 1. Понятие множества. 2. Описание множеств. 3. Операции над множествами: объединение, пересечение, разность, операции сравнения, операция in.	1						Устный опрос
2.4.12	Составление программ с использованием операций над множествами Составление программ с использованием операций объединения и пересечения множеств. Составление программ с использованием операции in.				2			Проверка лабораторной работы
2.4.13	Записи 1. Понятия записи, поля записи. 2. Описание записей. 3. Массивы записей. 4. Оператор with.	1						Устный опрос
2.4.14	Составление программ с использованием записей Составление программ с использованием отдельных записей. Составление программ с использованием массивов записей.					2		Проверка самостоятельной работы
2.4.15	Файлы 1. Понятие файлов. 2. Виды файлов. 3. Типизированные файлы. 4. Работа с типизированными файлами: создание файла, использование данных из файла, дополнение файла новыми данными.	2						Устный опрос
2.4.16	Составление программ с использованием файлов Составление программ записи данных в типизированный файл. Составление программ с использованием данных из типизированных файлов.				4			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
2.5	Основы программирования графики и звука	2			4			
2.5.1	<i>Основы программирования графики и звука</i> 1. Основы работы с графикой. Модуль Graph. Графические примитивы. Работа с цветом. Работа с пером и кистью. Действия со шрифтами. 2. Имитация движения графических объектов. 3. Построение графиков элементарных функций. 4. Деловая графика (столбчатые и круговые диаграммы). 5. Основы работы со звуком. Модуль Sounds. Процедуры и функции для работы со звуком.	2						Устный опрос
2.5.2	<i>Составление программ с использованием графики</i> Составление программ составления мозаики. Составление программ с движением графических объектов. Составление программ рисования графиков элементарных функций.				2			Проверка лабораторной работы
2.5.3	<i>Составление программ с использованием диаграмм</i> Составление программ рисования столбчатых диаграмм. Составление программ рисования круговых диаграмм.				1			Проверка лабораторной работы
2.5.4	<i>Составление программ с использованием звуковых эффектов</i> Составление программ с использованием кодов нот и включения звук. Составление программ с подключением готовых звуковых файлов.				1			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
3.	Основы современных технологий программирования	30			52			
3.1.	Парадигма объектно-ориентированного программирования. Платформа .NET	4						
3.1.1.	<i>Объектно-ориентированные языки и среды программирования</i> 1. Объектно-ориентированные языки. 2. Объектно-ориентированные среды программирования	1						Устный опрос
3.1.2.	<i>Философия и технология .NET.</i> 1. Компоненты платформы .NET. 2. Инфраструктура .NET Framework, строительные блоки .NET – CLR, CTS, CLS. 3. Базовые классы .NET.	2						Устный опрос
3.1.3.	<i>Модель выполнения приложений в среде .NET Framework.</i> 1. Промежуточный язык MSIL (Microsoft Intermediate Language). 2. Генерация управляемого кода. Сборка.	1						Устный опрос
3.2.	Основы объектно-ориентированного и компонентно-ориентированного языка программирования C#				8			
3.2.1.	<i>Синтаксис объектно-ориентированного языка C#.</i> Синтаксис языка. Встроенные типы и системные типы. Пространство имен System. Директива using System. Встроенная утилита Object Browser, просмотр содержимого пространства имен System.				1			Проверка лабораторной работы
3.2.2.	<i>Класс System.Console</i> Операции ввода и вывода. Чтение строки символов. Преобразование строки символов в число, метод Parse(). Формат вывода. Класс System.Math, его методы и константы.				1			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
3.2.3.	Операции и операторы языка программирования Операции, приоритеты операций. Операторы и выражения. Преобразование типов в выражениях.				2			Проверка лабораторной работы
3.2.4.	Одномерные массивы Одномерные массивы. Способы создания элементов массива. Класс System.Random, применение его методов для создания массива случайных чисел. Класс System.Array, его методы и свойство Length.				2			Проверка лабораторной работы
3.2.5.	Структуры Структуры. Создание структуры. Работа со структурами. Массивы структур.				2			Проверка лабораторной работы
3.3.	Объектно-ориентированное проектирование и программирование	10			8			
3.3.1.	Понятие объектно-ориентированного программирования. Объектные модели, принципы их построения, особенности реализации 1. Понятия объекта, объектной модели. 2. Понятие объектно-ориентированного программирования. 3. Характеристики объектно-ориентированного программирования.	1						Устный опрос
3.3.2.	Классы в C# 1. Форма определения класса. 2. Члены класса. 3. Спецификаторы доступа к членам класса.	1						Устный опрос
3.3.3.	Конструкторы 1. Параметризованные конструкторы. 2. Назначение оператора new. 3. Создание экземпляра класса.	1						Устный опрос

1	2	3	4	5	6	7	8	9
3.3.4.	Класс «Натуральные числа» Разработка класса «Натуральные числа». Определение членов-переменных класса. Разработка конструкторов и методов класса и их пошаговая отладка.				2			Проверка лабораторной работы
3.3.5.	Типы значений. Ссылочные типы 1. Динамическое использование памяти: стеки и кучи. 2. Сбор «мусора» и использование деструкторов.	1						Устный опрос
3.3.6.	Определение пользовательских методов класса 1. Спецификаторы доступа к методам класса. 2. Модификаторы ref, out, params параметров методов. 3. Возвращение методами объектов класса. 4. Возвращение методами массивов.	2						Устный опрос
3.3.7.	Использование модификаторов параметров методов Использование модификаторов ref, out, params параметров методов в классе «Натуральные числа».				2			Проверка лабораторной работы
3.3.8.	Общие (статические) члены класса 1. Доступ к общим членам класса. 2. Ограничения на static-методы.	2						Устный опрос
3.3.9.	Класс «Числовой массив» Разработка класса «Числовой массив». Определение статических и нестатических членов-переменных класса. Разработка статических и нестатических методов класса.				2			Проверка лабораторной работы
3.3.10.	Решение задач с использованием класса «Числовой массив Решение задач с использованием класса «Числовой массив». Разработка и тестирование методов класса, необходимых для решения задач.				2			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
3.3.11.	Цикл «жизни» переменных 1. Область видимости и время существования переменных. 2. Преобразование и приведение типов.	1						Устный опрос
3.3.12.	Исключительные ситуации Обработка исключительных ситуаций.	1						Устный опрос
3.4.	Принципы объектно-ориентированного программирования	12			24			
3.4.1.	Перегрузка методов и конструкторов как механизм реализации полиморфизма 1. Перегрузка методов. 2. Перегрузка конструкторов. 3. Понятие стека. 4. Класс «Стек из символов».	2						Устный опрос
3.4.2.	Класс «Стек» Класс «Стек» из данных встроенного типа, соответствующего заданию. Перегрузка конструкторов и методов класса.				2			Проверка лабораторной работы
3.4.3.	Перегрузка операторов (операций) 1. Перегрузка бинарных операций. 2. Перегрузка унарных операций и операций отношений.	2						Устный опрос
3.4.4.	Класс «Point» Разработка класса «Point». Перегрузка конструкторов, конструктор копий. Перегрузка арифметических операций (бинарных, унарных) и операций отношений для класса Point.				2			Проверка лабораторной работы
3.4.5.	Инкапсуляция и свойства Инкапсуляция и свойства. Формат записи свойств и правила их использования. Разработка свойств для полей класса Point и решение задач с их использованием.				2			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
3.4.6.	Индексаторы. Одномерные индексаторы 1. Определение одномерного индексатора. 2. Класс «Безопасный массив». 3. Виртуальный массив.	2						Устный опрос
3.4.7.	Одномерные индексаторы Одномерные индексаторы. Разработка одномерного индексатора для класса «Числовой массив», для класса «Стек».				2			Проверка лабораторной работы
3.4.8.	Наследование, базовые и производные классы 1. Доступы к членам базовых классов. 2. Конструкторы и наследование. 3. Виртуальные методы и их переопределение как механизм реализации полиморфизма. 4. Абстрактные классы, переопределение абстрактных методов.	4						Устный опрос
3.4.9.	Класс Figure Наследование. Базовый класс Figure и его производные классы: Rectangle, Triangle, Kwadrat, Circle.				2			Проверка лабораторной работы
3.4.10.	Проект «Планиметрия» Разработка проекта «Планиметрия». Базовый класс Figure и производный класс Point.				2			Проверка лабораторной работы
3.4.11.	Проект «Планиметрия» Разработка проекта «Планиметрия». Производный класс Otrezok. Решение задач планиметрии с использованием классов Point, Otrezok.				2			Проверка лабораторной работы
3.4.12.	Проект «Планиметрия» Разработка проекта «Планиметрия». Производный класс PrjamaJa. Решение задач планиметрии с использованием класса PrjamaJ.				2			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
3.4.13.	Проект «Планиметрия» Разработка проекта «Планиметрия». Производный класс Circle. Решение задач планиметрии с использованием класса Circle.				2			Проверка лабораторной работы
3.4.14.	Проект «Планиметрия» Разработка проекта «Планиметрия». Производный класс Triangle. Решение задач планиметрии с использованием класса Triangle .				4			Проверка лабораторной работы
3.4.15.	Интерфейсы и множественное наследование 1. Определение интерфейса. 2. Реализация интерфейсов. 3. Некоторые стандартные интерфейсы среды .NET Framework.	2						Устный опрос
3.4.16.	Проект «Наследование» Создание интерфейса ITop для классов проекта «Наследование» и его реализация в соответствующих классах. Реализация интерфейсов ICloneable, IComparable в классе Point и их использование.				2			Проверка лабораторной работы
3.5.	Принципы обобщенного программирования и их реализация	4			4			
3.5.1.	Понятие обобщенного программирования 1. Принципы обобщенного программирования (абстрагирование, иерархия, типизация).	1						Устный опрос
3.5.2.	Коллекции и наборы 1. Обзор коллекций. 2. Интерфейсы коллекций. 3. Классы коллекций общего назначения.	1						Устный опрос
3.5.3.	Методы работы с различными структурами данных Классы ArrayList, Queue, Stack, другие типы наборов данных из пространства имен System.Collections. Методы работы с различными структурами данных.				1			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
3.5.4.	Динамическая идентификация типов 1. Проверка типа с помощью ключевого слова is. 2. Использование операторов as, typeof.	1						Устный опрос
3.5.5.	Проект «Наследование» Динамическая идентификация типов для экземпляров классов проекта «Наследование»				1			Проверка лабораторной работы
3.5.6.	Обобщения 1. Обобщения. Обобщенные наборы, методы и классы.	1						
3.5.7.	Разработка обобщений Разработка обобщенных наборов, методов и классов.				2			Проверка лабораторной работы
3.6.	Основы компонентно-ориентированного программирования				8			
3.6.1.	Принципы работы Windows-приложений Делегаты. Назначение делегатов. События и сообщения, их обработка. Объявление и генерация событий. Реализация обработчиков событий.				1			Проверка лабораторной работы
3.6.2.	Разработка объектной модели приложения Графический интерфейс приложений. Технология Windows Forms. Пространство имен System.WinForms, класс Form. Добавление форм к проекту. Методы и события формы.				1			Проверка лабораторной работы
3.6.3.	Элементы управления Элементы управления. Создание обработчиков событий элементов управления.				2			Проверка лабораторной работы
3.6.4.	Компонент и компонентная модель Понятие компонента и компонентной модели. Разработка программных компонентов. Приложение-сервер, приложение-контейнер (клиент).				2			Проверка лабораторной работы

1	2	3	4	5	6	7	8	9
	Интерфейс System.ComponentModel.IComponent и класс System.ComponentModel.Component.							
3.6.5.	<i>Windows-приложения прикладного характера</i> Использование собственных программных компонентов в Windows-приложениях прикладного характера.				2			Проверка лабораторной работы
	Итого:	58			88	4		

РЕПОЗИТОРИЙ БГПУ

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Абрамов, В.Г. Введение в язык Паскаль: учебное пособие для студентов высших учебных заведений, обучающихся по специальности 010501 «Прикладная математика и информатика», направлению 010400 «Информационные технологии» / В.Г. Абрамов, Н.П. Трифонов, Г.Н. Трифонова. – Москва: КноРус, 2011. – 380 с.
2. Гросс, К. С# 2008 / К. Гросс. – С.-Петербург: БХВ-Петербург, 2009. — 551 с.
3. Ишкова, Э.А. С#. Начала программирования / Э. А. Ишкова. – Москва: Бином, 2013. – 333 с.
4. Орлов, С.А. Теория и практика языков программирования: учебник для вузов. Стандарт 3-го поколения / С.А. Орлов. – Санкт-Петербург: Питер, 2013. – 688 с.
5. Павловская, Т.А. С#. Программирование на языке высокого уровня / Т.А. Павловская. – Санкт-Петербург [и др.]: Питер: Мир книг, 2013. – 432 с.
6. Павловская, Т.А. Паскаль. Программирование на языке высокого уровня: учебник для высших учебных заведений по направлению подготовки дипломированных специалистов «Информатика и вычислительная техника» / Т.А. Павловская. – Санкт-Петербург: Лидер, 2010. – 460 с.
7. Павловский, А.И. Задачи по алгоритмизации и программированию на языке Pascal: практическое пособие для учителей информатики и учащихся общеобразовательных учреждений / А.И. Павловский, В.В. Стащенко, А.В. Ломакин. – Гродно: ГОИРО, 2009. – 62 с.
8. Пенкрат, В.В. Методы алгоритмизации: пособие для студентов высших учебных заведений, обучающихся по специальности 1-02 05 03 «Математика. Дополнительная специальность» (1-02 05 03-02 «Математика. Информатика») / В.В. Пенкрат. – Минск: БГПУ, 2012. – 107 с.
9. Севернева, Е.В. Основы алгоритмизации и программирования: учебно-методический комплекс для студентов высших учебных заведений / Е.В. Севернева, Н.М. Жалобкевич. – Минск: БГАТУ, 2009. – 112 с.
10. Семакин, И.Г. Основы алгоритмизации и программирования: учебник / И.Г. Семакин, А.П. Шестаков. – Москва: Академия, 2011. – 391 с.
11. Симанович, С.В. Информатика. Базовый курс: учебник для вузов. Стандарт 3-го поколения. / С.А. Симанов. – Санкт-Петербург: Питер, 2011. – 640 с.
12. Стиллмен, Э. Изучаем С# / Э. Стиллмен, Дж. Грин. – Санкт-Петербург: Мир книг, 2012. – 694 с.
13. Троелсен, Э. Язык программирования С# 2010 и платформа .NET 4 / Э. Троелсен. – Москва: Вильямс, 2011. – 1392 с.
14. Фаронов, В.В. Turbo Pascal: учебное пособие для студентов высших учебных заведений, обучающихся по направлению подготовки «Информатика и

вычислительная техника» / В.В. Фаронов. – Санкт-Петербург: Питер: Мир книг, 2012. – 366 с.

15. Шилдт, Г. С# 4.0: полное руководство / Г. Шилдт. – Москва: Вильямс, 2012. – 1056 с.

Дополнительная литература

16. Буславский, А.А. Начальный уровень обучения программированию на языке Pascal / А.А. Буславский. – Минск: МОИРО, 2010. – 89 с.

17. Вучэбна-метадычны дапаможнік для студэнтаў вышэйшых навучальных устаноў / Г.А. Расолька, Ю.А. Крэмень. – Мінск: БДУ, 2008. – 391 с.

18. Голдобина, Т.А. Выполнение инженерных расчетов с применением прикладных программ: учебно-методическое пособие / Т.А. Голдобина, С.В. Дрючкова, Н.И. Чурак. – Гомель: БелГУТ, 2013. – 54 с.

19. Гросс, К. С# 2008 и платформа NET 3.5 Framework: вводный курс / К. Гросс. – Москва: Вильямс, 2009. – 466 с.

20. Гусева, А.И. Учимся программировать: Pascal 7.0. Задачи и методы их решения: учебное пособие для студентов высших учебных заведений / А.И. Гусева. – Москва: Диалог-МИФИ, 2011. – 215 с.

21. Кричевцов, О.В. Лабораторные работы по дисциплине «Основы алгоритмизации и программирования». Язык Pascal / О.В. Кричевцов. – Витебск: ВГТК, 2010. – 133 с.

22. Культин, Н.Б. Программирование в Turbo Pascal 7.0 и Delphi / Н.Б. Культин. – Санкт-Петербург: БХВ-Петербург, 2012. – 380 с.

23. Меженный, О.А. Turbo Pascal: самоучитель / О. А. Меженный. – Москва: Диалектика, 2008. – 335 с.

24. Москаленко, А.А. Решение прикладных задач в интегрированной среде Турбо Паскаль: методическое пособие / А.А. Москаленко, З.И. Кононенко. – Минск: БНТУ, 2011. – 60 с.

25. Нейгель, К. С# 2008 и платформа .NET 3.5 для профессионалов / К. Нейгел [и др.]. – Москва: Диалектика, 2009. – 1738 с.

26. Немнюгин, С.А. Turbo Pascal: программирование на языке высокого уровня: для студентов / С. А. Немнюгин. – Санкт-Петербург: Питер: Питер-Пресс, 2008. – 543 с.

27. Нортроп, Т. Основы разработки приложений на платформе Microsoft .NET Framework / Т. Нортроп, Ш. Уилдермьюс, Б. Райан. – Москва: Русская редакция, Санкт-Петербург: Питер, 2007. – 864 с.

28. Павловский, А.И. Теоретические основы алгоритмизации. / А.И. Павловский, В.В. Пенкрат – Минск: БГПУ, 2007. – 59 с.

29. Потапова, Л.Е. Алгоритмизация и программирование на языке Паскаль: учебно-методическое пособие / Л.Е. Потапова, Т.Г. Алейникова. – Витебск: ВГУ, 2008. – 129 с.

30. Пшеничнов, Ю.А. Информатика: лабораторный практикум для студентов технических специальностей / Ю.А. Пшеничнов. – Гомель: БелГУТ,

2011. — 47 с.

31. Пянкрат, В.У. ІнТАл і Pascal у паралельным выкладанні / В.У. Пянкрат – Мінск: БДПУ, 2008.– 74 с.

32. Расолька, Г.А. Заданні вылічальнай практыкі па курсу «Метады праграміравання і інфарматыка»: дапаможнік для студэнтаў механіка-матэматычнага факультэта спецыяльнасці 1-31 03 01-02 «Матэматыка (навукова-педагагічная дзейнасць)» / Г.А. Расолька, Е.В. Крэмень, Ю.А. Крэмень. – Мінск: БДУ, 2013. – 111 с.

33. Расолька, Г.А. Паскаль: тэорыя і практыка праграміравання: вучэбна-метдычны дапаможнік для студэнтаў вышэйшых навучальных устаноў / Г.А. Расолька, Ю.А. Крэмень. – Мінск: БДУ, 2008. – 391 с.

34. Расолька, Г.А. Система тестов для самоподготовки по курсу «Методы программирования и информатика». Язык Pascal: пособие для студентов механико-математического факультета специальности 1-31 03 01-02 «Математика (научно-педагогическая деятельность)» / Г.А. Расолька, Е.В. Кремень, Ю.А. Кремень. – Минск: БГУ, 2013. – 70 с.

35. Романькова, Т. Л. Объектно-ориентированное программирование: курс лекций / Т.Л. Романькова. – Гомель: ГГТУ, 2010. – 97 с.

36. Силаев, Н.В. Методы решения задач информатики: пособие для студентов математического факультета специальностей 1-02 05 05-01 Информатика. Иностранный язык (английский), 1-02 05 03-02 Математика и информатика, 1-02 05 03 Математика / Н. В. Силаев, З. Н. Силаева. – Брест: БрГУ, 2011. – 62 с.

37. Скит, Д. С#. Программирование для профессионалов / Джон Скит – Москва: Вильямс, 2011. – 544 с.

38. Фонасов, С.А. Числовые задачи в программировании: сборник / С.А. Фонасов. – Гродно: Гродненский областной институт развития образования, 2012. – 66 с.

Методические рекомендации по организации и выполнению самостоятельной работы студентов

Содержание и формы контролируемой самостоятельной работы студентов разрабатываются кафедрами в соответствии с целями и задачами подготовки специалиста. Для управления самостоятельной работой рекомендуется использовать электронные средства обучения, тестирующие программы. Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

Перечень используемых средств диагностики результатов учебной деятельности

Основным средством диагностики усвоения знаний, умений и овладения необходимыми навыками по учебной дисциплине являются:

– *фронтальный опрос* на лекционных занятиях, направлен систематизацию знаний студентов, определение уровня готовности аудитории к восприятию нового материала, а также на формирование у преподавателя представление об усвоении студентами основополагающих понятий и фактов изучаемой учебной дисциплины;

– *проверка заданий* разнообразного типа (рецептивные, репродуктивные, продуктивные, творческие), выполняемых в рамках часов, отводимых на учебные занятия (лабораторные), представляет собой диагностику систематичности подготовки студентов к занятиям, уровень усвоения ими практико-ориентированного содержания программного материала учебной дисциплины;

– *групповые и индивидуальные консультации студентов* предназначены для диагностики уровня овладения определенными знаниями, умениями и навыками, как теоретического материала, так и практического; устранения типичных ошибок и пробелов в знаниях обучающихся;

– *самостоятельные работы* используются для определения индивидуальных особенностей, темпа продвижения студентов и усвоения ими необходимых знаний;

– *компьютерное тестирование* позволяет относительно быстро провести диагностику усвоения студентами учебного материала как по отдельным темам и разделам учебной дисциплины, так и по учебной дисциплине в целом;

– *зачет* используется для осуществления итоговой диагностики усвоения учащимися содержания учебной дисциплины за учебный семестр и оценивается обычно в форме «зачтено» или «не зачтено» в соответствии с критериями оценки результатов учебной деятельности обучающихся в учреждениях высшего образования по десятибалльной шкале;

– *экзамен* используется для осуществления итоговой диагностики усвоения учащимися содержания по учебной дисциплины и оценивается в соответствии с критериями оценки результатов учебной деятельности обучающихся в учреждениях высшего образования по десятибалльной шкале.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Введение в математику	Кафедра алгебры и геометрии	В теме 2.1 «Основные элементы языка программирования» при изложении материала о логических операциях и логических выражениях использовать согласованную терминологию.	Протокол № 2 от 18.09.2014
Элементарная математика и практикум по решению задач	Кафедра математики и методики преподавания математики	В лабораторных работах «Составление программ с использованием алгоритмической конструкции «Следование»»; «Составление программ с использованием алгоритмической конструкции «Ветвление»»; «Составление программ с использованием алгоритмической конструкции «Повторение»» предусмотреть задания на совершенствование знаний основных математических формул.	Протокол № 2 от 18.09.2014